

Docket No. AUS000147US1

**METHOD AND APPARATUS FOR UPDATING BOOT CODE IN A DATA
PROCESSING SYSTEM ON A LOCAL STORAGE DEVICE**

CROSS REFERENCE TO RELATED APPLICATIONS

5

The present invention is related to applications
entitled AUTOMATIC RECOVERY OF A CORRUPTED BOOT IMAGE IN
A DATA PROCESSING SYSTEM, serial no. 09/527397, attorney
docket no. AUS000145US1; METHOD AND APPARATUS FOR COPYING
10 A BOOTABLE IMAGE FROM A NETWORK TO A LOCAL BOOT DEVICE IN
A DATA PROCESSING SYSTEM, serial no. 09/527399, attorney
docket no. AUS000148US1; all of which are filed even date
hereof, assigned to the same assignee, and incorporated
herein by reference.

KS
10/24/02

KS
10/24/02

15

BACKGROUND OF THE INVENTION

1. Technical Field:

The present invention relates generally to an
20 improved data processing system and in particular to a
method and apparatus for updating code in a data
processing system. Still more particularly, the present
invention relates to a method and apparatus for updating
boot code in a data processing system.

25

2. Description of Related Art:

When a computer is started or reset, instructions
executed by a processor are used to load and start the
operating system for the computer. These instructions
30 are also referred to as "boot code". Typically, this
function may be provided within a basic input/output

Docket No. AUS000147US1

system (BIOS). A BIOS provides a set of essential software routines that test the hardware at startup, start the operating system, and transfer the supportive data among hardware devices. Typically, BIOS's may be
5 stored in a read-only memory (ROM) so that the BIOS may be executed when the computer is turned on. Additionally, the BIOS may be stored on some other persistent storage device in which the BIOS may be updated, such as a flash memory. In booting the
10 operating system, the instructions search for the operating system, load the operating system, and then pass control to the operating system.

Updates to operating systems often may require updates to the code used to boot the operating system.
15 This code is also referred to as the "boot code". In some instances, if the operating system is updated to another version, a particular version of the boot code also may be required. If the boot code remains un-updated, the operating system may not be properly
20 loaded by the boot code. In some cases, the operating system may not run at all. With frequent changes to operating systems, it may be difficult to ensure that the version of the boot code is the correct version required by the operating system.

25 Therefore, it would be advantageous to have an approved method and apparatus for maintaining a correct version of boot code for an operating system.

Docket No. AUS000147US1

SUMMARY OF THE INVENTION

The present invention provides a method and apparatus for updating a current boot code in a data processing system in which the current boot code is used to load an operating system from a storage device. The storage device is searched for an updated boot code for the operating system in response to starting the data processing system. The current boot code is updated prior to loading the operating system for the data processing system if the updated boot code is present in the storage device.

Docket No. AUS000147US1

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the
5 invention are set forth in the appended claims. The
invention itself, however, as well as a preferred mode of
use, further objectives and advantages thereof, will best
be understood by reference to the following detailed
description of an illustrative embodiment when read in
10 conjunction with the accompanying drawings, wherein:

Figure 1 is a pictorial representation of a data
processing system in which the present invention may be
implemented in accordance with a preferred embodiment of
the present invention;

15 **Figure 2** is a block diagram of a data processing
system shown in which the present invention may be
implemented;

Figure 3 is a block diagram of a boot code image in
accordance with a preferred embodiment of the present
20 invention; and

Figure 4 is a flowchart of a process for booting a
computer in accordance with a preferred embodiment of the
present invention.

Docket No. AUS000147US1

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

With reference now to the figures and in particular with reference to **Figure 1**, a pictorial representation of a data processing system in which the present invention may be implemented is depicted in accordance with a preferred embodiment of the present invention. A computer **100** is depicted which includes a system unit **110**, a video display terminal **102**, a keyboard **104**, storage devices **108**, which may include floppy drives and other types of permanent and removable storage media, and mouse **106**. Additional input devices may be included with personal computer **100**, such as, for example, a joystick, touchpad, touch screen, trackball, microphone, and the like. Computer **100** can be implemented using any suitable computer, such as an IBM RS/6000 computer or IntelliStation computer, which are products of International Business Machines Corporation, located in Armonk, New York. Although the depicted representation shows a computer, other embodiments of the present invention may be implemented in other types of data processing systems, such as a network computer. Computer **100** also preferably includes a graphical user interface that may be implemented by means of systems software residing in computer readable media in operation within computer **100**.

With reference now to **Figure 2**, a block diagram of a data processing system is shown in which the present invention may be implemented. Data processing system **200** is an example of a computer, such as computer **100** in

Docket No. AUS000147US1

Figure 1, in which code or instructions implementing the processes of the present invention may be located. Data processing system **200** employs a peripheral component interconnect (PCI) local bus architecture. Although the depicted example employs a PCI bus, other bus architectures such as Accelerated Graphics Port (AGP) and Industry Standard Architecture (ISA) may be used. Processor **202** and main memory **204** are connected to PCI local bus **206** through PCI bridge **208**. PCI bridge **208** also may include an integrated memory controller and cache memory for processor **202**. Non-volatile memory **209** also is connected to PCI local bus **206** and, in this example, contains a BIOS, which includes a boot code. In this example, non-volatile memory **209** may take the form of a non-volatile random access memory or an erasable programmable read only memory (EPROM). In the depicted examples, the operating system also may be located in non-volatile memory **209**.

Additional connections to PCI local bus **206** may be made through direct component interconnection or through add-in boards. In the depicted example, local area network (LAN) adapter **210**, small computer system interface SCSI host bus adapter **212**, and expansion bus interface **214** are connected to PCI local bus **206** by direct component connection. In contrast, audio adapter **216**, and graphics adapter **218** are connected to PCI local bus **206** by add-in boards inserted into expansion slots. Expansion bus interface **214** provides a connection for a keyboard and mouse adapter **220**, modem **222**, and local storage device **224**. Local storage device **224**, in this example, contains

Docket No. AUS000147US1

an image of the operating system. The boot code in non-volatile memory **209** will boot the operating system image located in local storage device **224**. Local storage device **224** may take various forms, such as a flash memory.

5 Flash memory is a memory chip than can be rewritten and hold its content without power. Flash memory is a type of non-volatile memory. Flash memory may take various forms, such as a memory stick, which is a flash memory card designed for various devices. These memories typically

10 vary from 4 MB to 192 MB, but may come in larger sizes. Of course, local storage device **224** may take other forms, such as, for example, a floppy disk drive, a CD-ROM, or a read-only memory. SCSI host bus adapter **212** provides a connection for hard disk drive **226**, tape drive **228**, and

15 CD-ROM drive **230**. Typical PCI local bus implementations will support three or four PCI expansion slots or add-in connectors.

An operating system runs on processor **202** and is used to coordinate and provide control of various components

20 within data processing system **200** in **Figure 2**. The operating system may be a commercially available operating system such as Windows 2000, which is available from Microsoft Corporation. Instructions for the operating system and applications or programs are located on storage

25 devices, such as hard disk drive **226** and local storage device **224**, and may be loaded into main memory **204** for execution by processor **202**.

Those of ordinary skill in the art will appreciate that the hardware in **Figure 2** may vary depending on the

30 implementation. Other internal hardware or peripheral

Docket No. AUS000147US1

devices, such as flash ROM (or equivalent non-volatile memory) or optical disk drives and the like, may be used in addition to or in place of the hardware depicted in **Figure 2**. Also, the processes of the present invention
5 may be applied to a multiprocessor data processing system.

For example, data processing system **200**, if optionally configured as a network computer, may not include SCSI host bus adapter **212**, hard disk drive **226**,
10 tape drive **228**, and CD-ROM **230**, as noted by dotted line **232** in **Figure 2** denoting optional inclusion. In that case, the computer, to be properly called a client computer, must include some type of network communication interface, such as LAN adapter **210**, modem **222**, or the
15 like. As another example, data processing system **200** may be a stand-alone system configured to be bootable without relying on some type of network communication interface, whether or not data processing system **200** comprises some type of network communication interface. As a further
20 example, data processing system **200** may be a personal digital assistant (PDA) device, which is configured with ROM and/or flash memory in order to provide non-volatile memory for storing operating system files and/or user-generated data.

25 The depicted example in **Figure 2** and above-described examples are not meant to imply architectural limitations. For example, data processing system **200** also may be a notebook computer or hand held computer in addition to taking the form of a PDA. Data processing
30 system **200** also may be a kiosk or a Web appliance.

The present invention provides a method, apparatus,

Docket No. AUS000147US1

and computer implemented instructions for updating the boot code. The mechanism of the present invention is initiated when a computer is started. The boot code will search a local storage device, such as local storage
5 device **224**, for the new boot code image. If the boot code image is found and this image is a later version than the one that is currently being used, the boot code will automatically update to the newer version, reboot, and then load and run the operating system.

10 In the depicted examples, the appropriate version of the boot code is stored in conjunction with the operating system in the same storage device. This version of the boot code is one that will properly boot or load the operating system. In these examples, a boot code image
15 and the operating system are stored in local storage device **224**. When the boot code currently being used in data processing system **200** is initiated, this boot code will search for a presence of an update boot code image in local storage device **224**. If an update boot code
20 image is found, the update boot code image in local storage device **224** is used to update the current boot code. Alternatively, the version of this image in local storage device **224** will be compared to that of the boot code in non-volatile memory **209**. If the boot code
25 currently being used is of the correct version, the system will continue to boot. Otherwise, an update will be made and the system will be restarted.

Of course, depending on the implementation, the update boot code image could be located in another
30 location. For example, the current boot code image could be obtained through a network connection provided by LAN

Docket No. AUS000147US1

adapter **210** to a server or some other device.

In the depicted examples, the correct version of the boot code is located in the same storage device as the operating system. With this configuration, the boot code
5 on the computer can always search for the boot file or image on the same storage device as the operating system.

Turning now to **Figure 3**, a block diagram of a boot code image is depicted in accordance with a preferred embodiment of the present invention. Boot code image **300**
10 includes a header **302** and boot code **304**. Boot code **304** is an update boot code, which may be used to update the current boot code located in the non-volatile memory in the data processing system. In these examples, the update is a replacement of the current boot code with the
15 update boot code. Of course, other types of updates may occur, such as replacing portions of the current boot code with the update boot code.

A check may be made to see whether boot code **304** should replace the current boot code. For example, a
20 check may be made to see if boot code **304** is a newer version of the current boot code in these examples. This check is made, in this example, by examining information within header **302**. Header **302**, in this example, contains identification (ID) information **306** and error checking
25 (EC) information **308**. ID information **306** may contain, for example, a version number or other information used to determine whether to replace the current boot code with boot code **304**.

Header **302** also may contain information to verify
30 whether boot code image **300** is corrupted or contains errors. For example, checksum information or cyclical

Docket No. AUS000147US1

redundancy checking (CRC) information may be used to determine whether boot code image **300** is a good image or a corrupted image. A checksum is a calculated value used to test data for the presence of errors that can occur
5 when data is transmitted or when it is written to a disk. The checksum is calculated for a chunk of data by sequentially combining all of the bytes of data with a series of arithmetic or logical operations. After the data is transmitted or stored, the new checksum is
10 calculated in the same way using the transmitted or stored data. If the two checksums do not match, then an error has occurred and the data should be transmitted or stored again. CRC involves using a calculation to generate a number based on the data transmitted. The
15 sending device performs the calculation before transmission and sends the result to a receiving device. The receiving device repeats the same calculation after transmission. If the result is the same, then the transmission is assumed to be error free. Of course,
20 other mechanisms may be used to determine whether boot code **304** is error free. Of course, other mechanisms may be used to determine whether errors are present in the image.

With reference now to **Figure 4**, a flowchart of a
25 process for booting a computer is depicted in accordance with a preferred embodiment of the present invention. The steps in **Figure 4** form a process used in boot code to form updates on a local storage device. This process may be implemented in a BIOS for a computer.

30 The process begins by initializing hardware (step **400**). This initialization of hardware includes steps,

Docket No. AUS000147US1

such as, for example, searching for other BIOS's in plug-in boards that may be connected to the computer, setting up pointers, such as interrupt vectors, and memory to access those routines. A search is then made
5 for the update boot code image (step 402). This image may be, for example, boot code image 300 in Figure 3. This search is typically made on the same device on which the operating system for the computer is located. Of course, the search may be to other locations, such as
10 another local storage device or even a remote storage device, such as a remote hard disk or server in communication with the computer.

A determination is made as to whether an update boot code image is present (step 404). This update boot code
15 may be, for example, boot code image 300 in Figure 3. In this example, if the update boot code image is present, a replacement will be made. Of course, in addition to determining whether an image is present, this step may include a comparison of the update boot code image with
20 the current boot code image to determine whether an update should be made. This identification may be made in a variety of ways. For example, if the boot code image includes version information, this version information may be placed in a header in an image or file
25 containing the boot code. The version information may be extracted from the header and compared to the version number for the current boot code to see whether the current version is the correct version. If the update boot code image is to be used to update the current boot
30 code, then the result of step 404 indicates that an update boot code image has been present.

Docket No. AUS000147US1

If another update boot code image is present, a check of this update boot code image is made to see if the update boot code image is a good image (step 406). This step is used to determine whether the update boot
5 code image has errors or is corrupted. Step 406 may be performed using error checking information located in a header, such as EC information 308 which is for the update boot code image in **Figure 3**.

If the current image is not a good image, then the
10 boot code is updated (step 408). Step 408 involves copying the update boot code image from the local storage into the flash memory in which the boot code is located on the computer. Thereafter, the process returns to step 400 to begin the initialization process again.

15 With reference back to step 404, if an update boot code image is absent, the operating system is booted (step 410) with the process terminating thereafter. With reference again to step 406, if the update boot code image is a good update boot code image, the process
20 proceeds to step 410 to boot the operating system.

Thus, the present invention provides a method, apparatus, and computer implemented instructions for maintaining a correct version of boot code for use in booting an operating system. In one implementation, a
25 local storage device, such as a flash memory, contains both the operating system and the update boot code image. In such a case, the handling of the operating system in the boot code enables access to a required version of the boot code at all times. Thus, whenever a new version of
30 an operating system is introduced, the appropriate version of the boot code also may be included in the same

Docket No. AUS000147US1

local storage device. Alternatively, the boot code image may be located in a remote storage device or in some other local storage device. With a remote storage device, the current boot code may search a remote
5 location to see whether updates have occurred for the boot code. In such a case, the computer must be initialized to a point such that the communications adapter in the computer is able to establish a data connection to allow a search for an updated boot code
10 image.

It is important to note that while the present invention has been described in the context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes of
15 the present invention are capable of being distributed in the form of a computer readable medium of instructions and a variety of forms and that the present invention applies equally regardless of the particular type of signal bearing media actually used to carry out the
20 distribution. Examples of computer readable media include recordable-type media, such as a floppy disk, a hard disk drive, a RAM, CD-ROMs, DVD-ROM's, and transmission-type media, such as digital and analog communications links, wired or wireless communications
25 links using transmission forms, such as, for example, radio frequency and light wave transmissions. The computer readable media may take the form of coded formats that are decoded for actual use in a particular data processing system.

30 The description of the present invention has been presented for purposes of illustration and description,

Docket No. AUS000147US1

and is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art. For example, although the depicted examples

5 place the update boot code image in the same location as the operating system, these two components may be placed in different locations. For example, the update boot code image may actually be located at a particular location, such as on a server. In this type of

10 implementation, the mechanism of the present invention would look to that location to see whether an update boot code image is present. The embodiment was chosen and described in order to best explain the principles of the invention, the practical application, and to enable

15 others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated.

00527390-034600